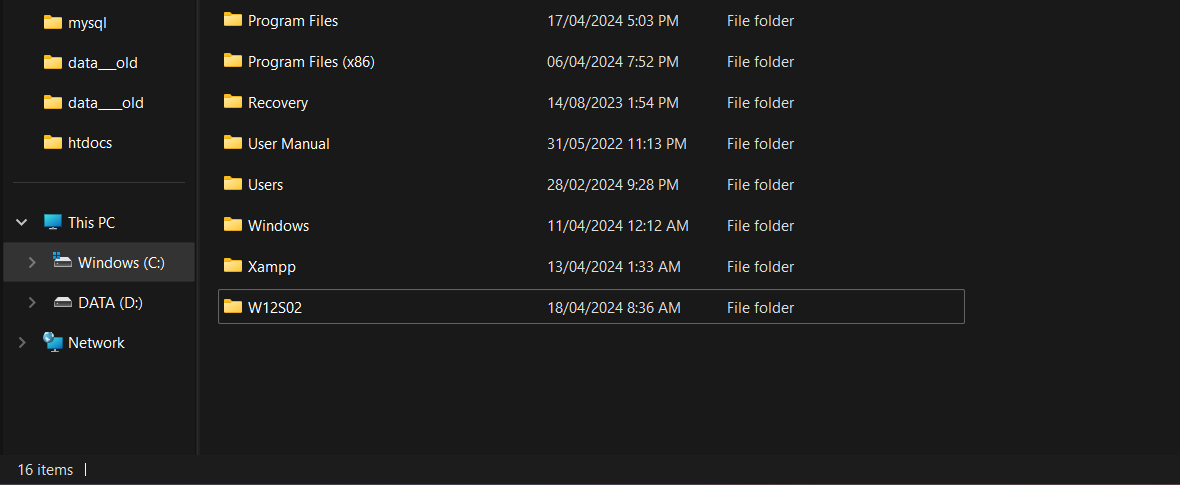
|  |
| --- |
| **­­­­­­­­LAPORAN PRAKTIKUM PENGENALAN APLIKASI BERBASIS INTERNET**  **CRUD LARAVEL** |
| ­­­­ |
| **Agus Pranata Marpaung**  **13323033**  **DIII TEKNOLOGI KOMPUTER** |
| **INSTITUT TEKNOLOGI DEL**  **FAKULTAS VOKASI** |

**Judul Praktikum**

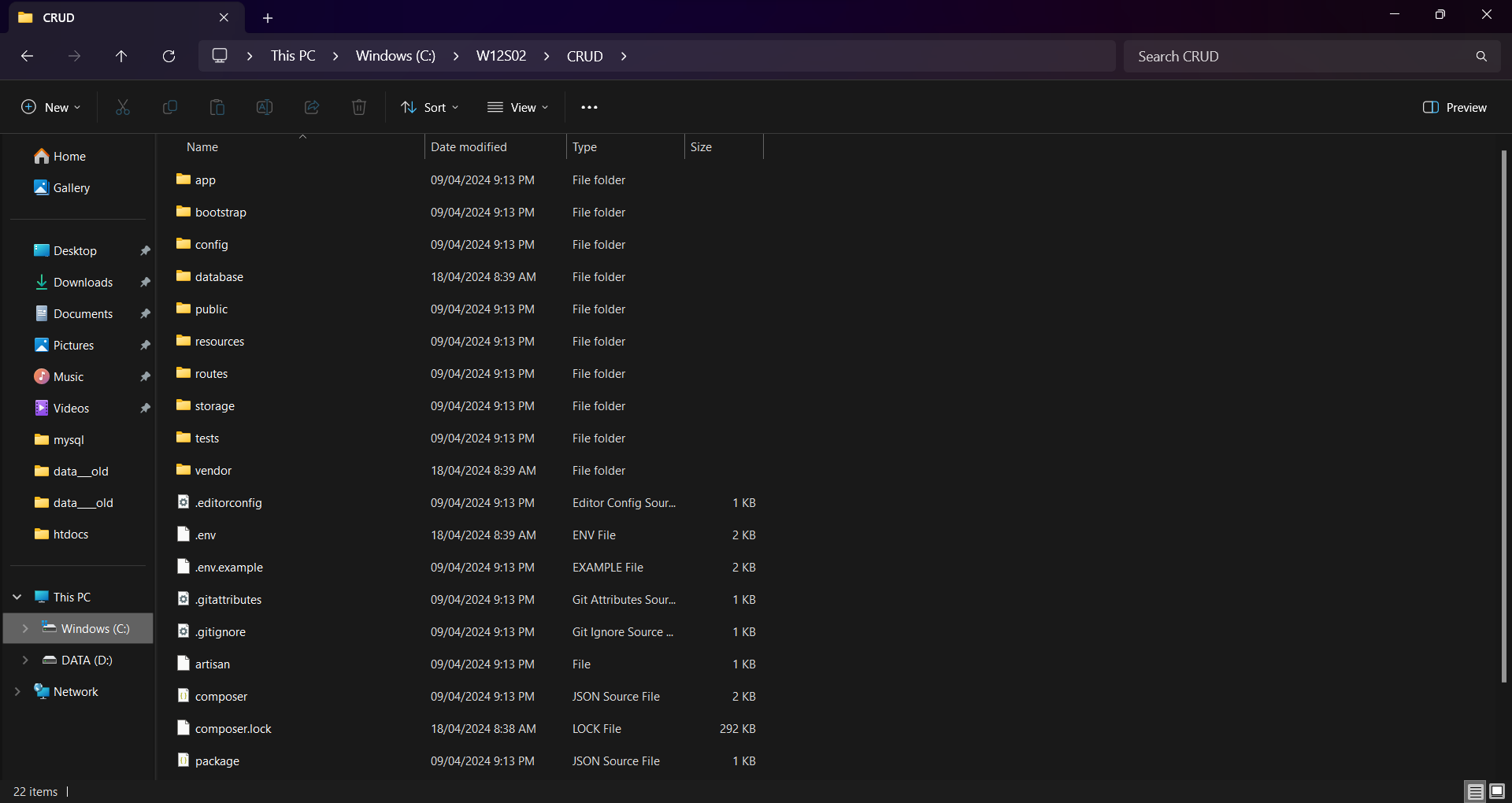
|  |  |  |
| --- | --- | --- |
| **Minggu/Sesi** | : | XII/2 |
| **Kode Mata Kuliah** | : | 1331204 |
| **Nama Mata Kuliah** | : | PENGENALAN APLIKASI BERBASIS INTERNET |
| **Setoran** | : | *Softcopy* |
| **Batas Waktu Setoran** | : | Akhir sesi |
| **Tujuan** | : | 1. *Mahasiswa dapat melakukan praktikum mengenai CRUD pada laravel 11* |

Membuat CRUD dengan Laravel

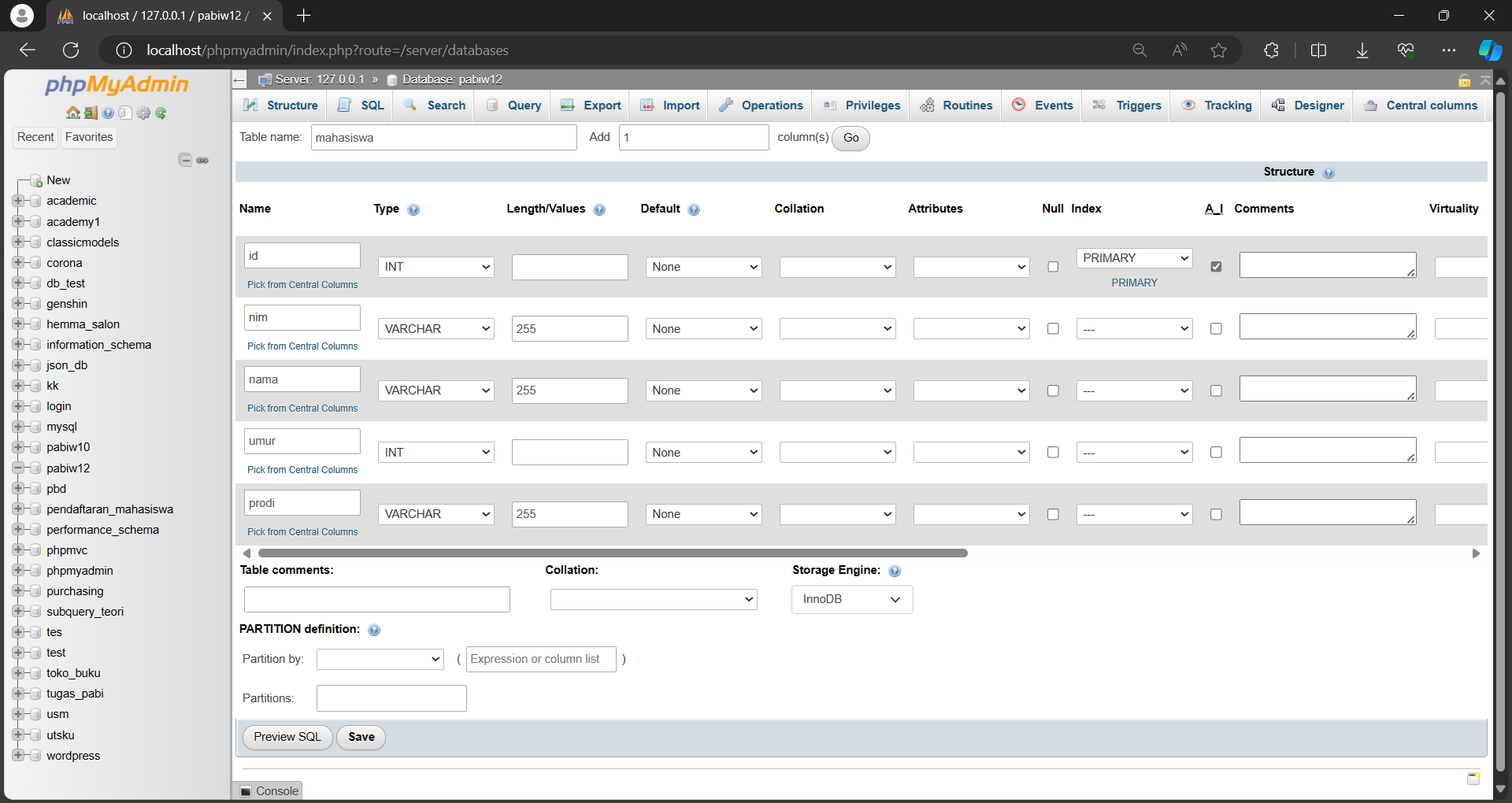
1. Buatlah Folder baru dengan nama **W12S02**, kemudian lakukan instalasi Laravel dengan nama project **CRUD** menggunakan composer.



Berikut folder hasil instalasi laravel yang dilakukan:



1. Kemudian aktifkan **Apache** dan **MySQL** melalui **XAMPP**, dan tambahkan database baru dengan nama **pabiw12.** Buat tabel **mahasiswa** dengan ketentuan sebagai berikut “id” dibuat auto increment.



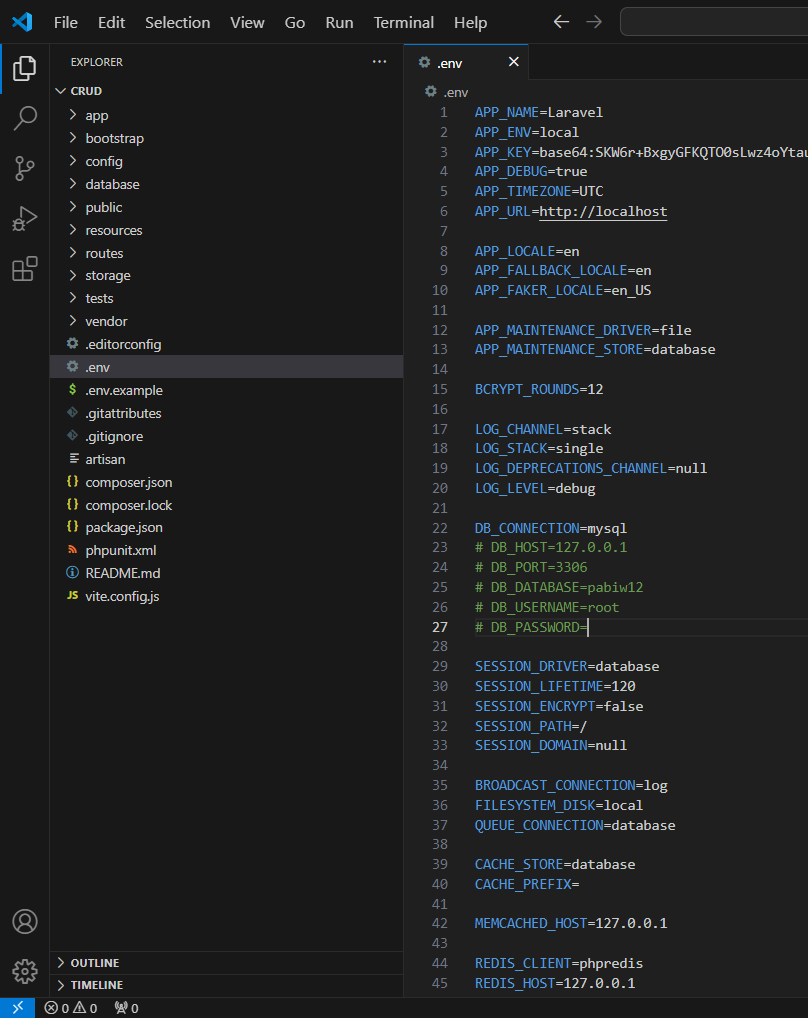
Tambahkan beberapa data pada tabel tersebut, anda dapat menggunakan data sebagai berikut.

insert into mahasiswa values

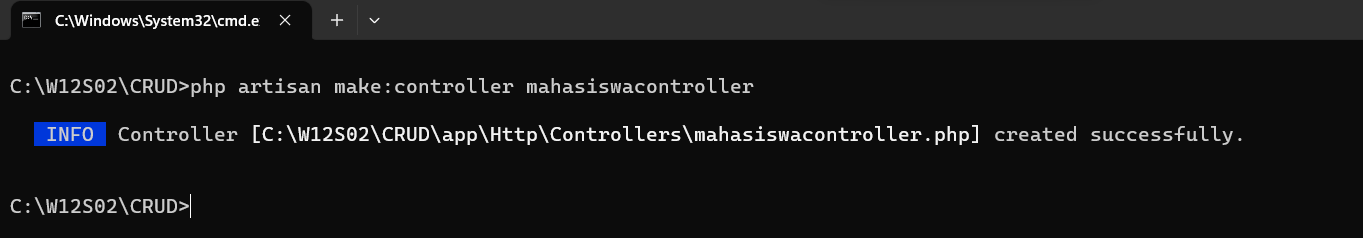
(null,'1333001','Anya Forger','18','D3 Teknologi Komputer'),

(null,'1333002','Damian Desmond','17','D3 Teknologi Komputer'), (null,'1333003','Becky Blackbell','19','D3 Teknologi Komputer')

1. Kemudian lakukan edit pada file **.env** menggunakan editor sesuaikan dengan database yang telah dibuat sebelumnya.

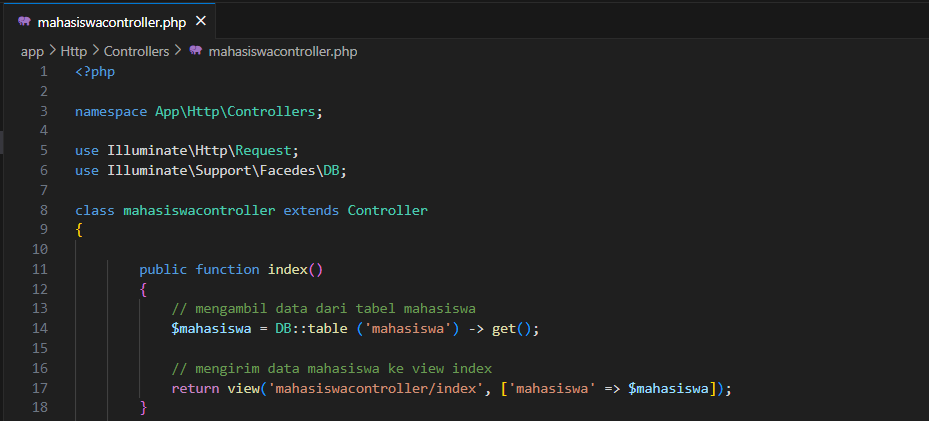


1. Pada folder CRUD tambahkan controller baru menggunakan command berikut pada cmd.



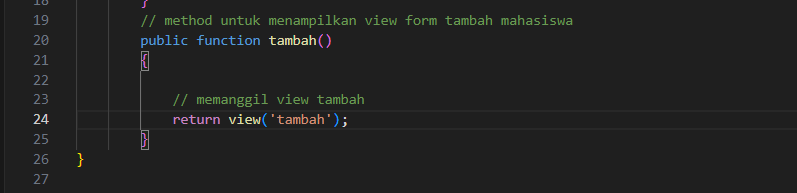
Controller digunakan untuk menghubungkan antara Model dengan View dan biasanya controller akan berisi logika-logika dari aplikasi yang akan dibangun. Jika perintah berhasil dijalankan akan mendapatkan file controller baru yang berada di dalam folder app\Http\Controllers\mahasiswacontroller.php.

1. Kemudian lakukan edit pada mahasiswacontroller.php sehingga menjadi seperti berikut.



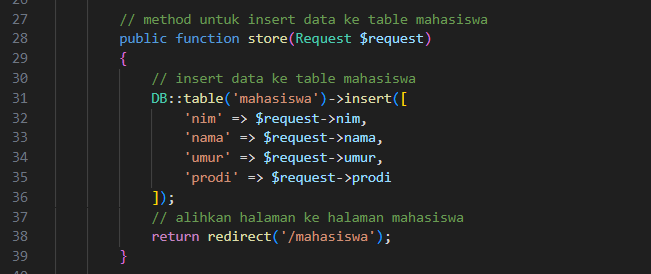
Penjelasan:

* + Fungsi index, akan diambil data dari table ‘mahasiswa’ dengan perintah $mahasiswa = DB::table('mahasiswa')->get();
  + Line 13: Fungsi table() kita menentukan nama table yang ingin dipilih. Fungsi get() berguna untuk mengambil data dari table yang dipilih. - Line 16: Passing data atau kirim data ke view agar ditampilkan.



Penjelasan:

Method tambah() berfungsi untuk menampilkan view tambah, yang merupakan form penginputan data mahasiswa.

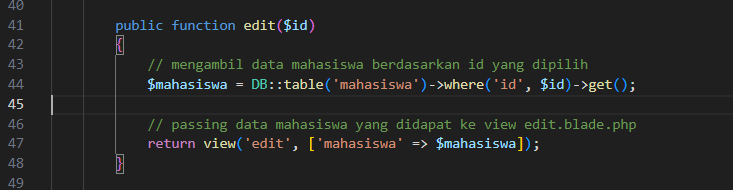


Penjelasan:

Method store digunakan untuk menangkap data yang dikirimkan dari form tambah pegawai. Cara menangkap datanya di dalam parameter fungsi store(). Diberi keterangan **Request** diikuti dengan nama variabel yang ingin kita jadikan sebagai penyimpan data yang dikirim dari form **$request.**

Kemudian insert datanya ke tabel mahasiswa dengan menggunakan query builder laravel untuk insert data. Fungsi tabel() untuk memberitahukan nama tabel. Fungsi insert() bertujuan untuk menginsert data dan menetapkan data apa saja yang ingin di insert.

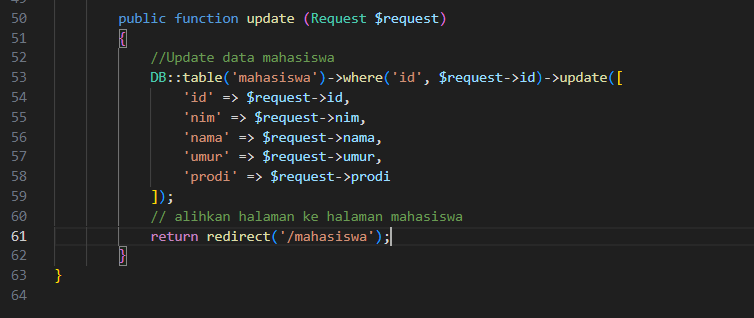
Kemudian setelah kita menginput data ke tabel mahasiswa, maka diberikan perintah untuk mengalihkan halaman ke halaman mahasiswa atau route ‘mahasiswa’ kembali.



Penjelasan:

Parameter method edit berfungsi untuk menangkap data id yang dikirimkan dari route ‘/mahasiswa/edit/id’.

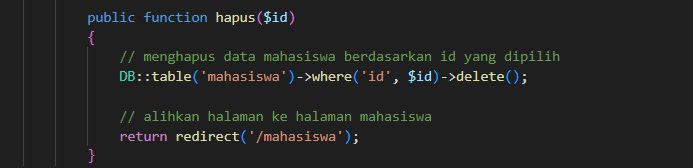
Selanjutnya akan diambil data mahasiswa dari database sekalian menseleksi data id yang ingin di ambil dengan menyeleksi id mahasiswa. Kemudian passing data ke view edit.blade.php



Penjelasan:

Seperti pada saat membuat proses penginputan data ke database, data yang dikirim dari form ditangkap dengan mendeklarasikan Request dalam parameter method update.

Data-data yang dikirim dari form edit akan tersimpan didalam variabel $request. Untuk update data digunakan query builder (line 54-59). Kemudian dialihkan kembali halamannya ke route ‘/mahasiswa’ yang menampilkan semua data mahasiswa.



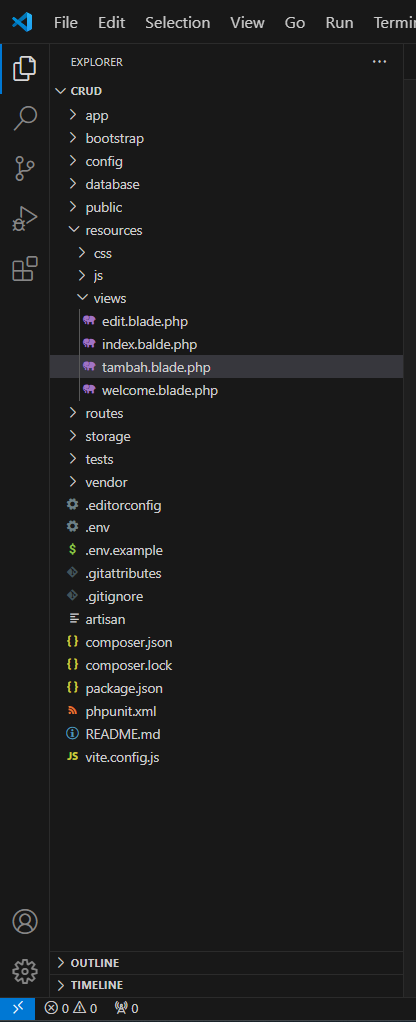
Penjelasan:

Pada method hapus, data id akan ditangkap dalam parameter method hapus, sama seperti kita menangkap data id pada method edit.

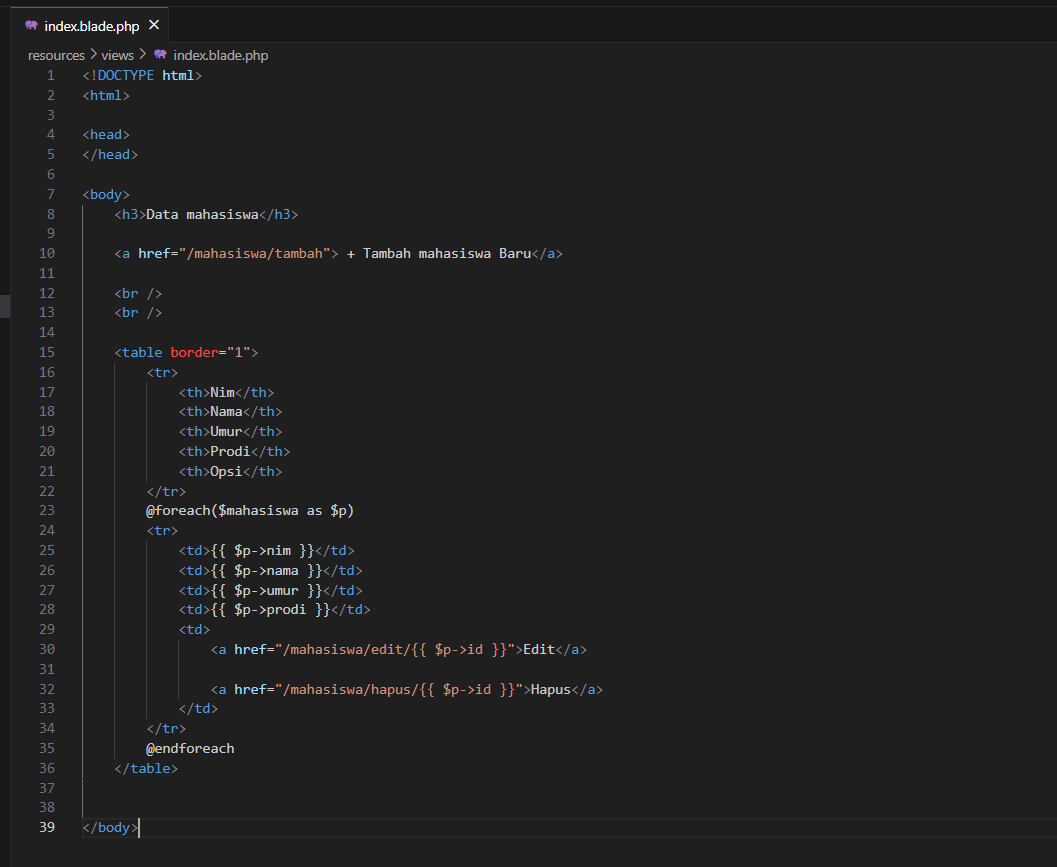
Kemudian dibuat query builder untuk menghapus data dari tabel mahasiswa. Pilih tabel dalam fungsi tabel(), kemudian seleksi dengan fungsi where() untuk menentukan data mana yang ingin dihapus. Lalu gunakan fungsi delete() untuk menghapus datanya().

Setelah penghapusan data, diberikan perintah untuk mengalihkan halaman nya kembali ke route ‘/mahasiswa’.

1. Kemudian pada folder CRUD/resources/views tambahkan 3 file baru dengan nama **index.blade.php** , **tambah.blade.php** , dan **edit.blade.php**



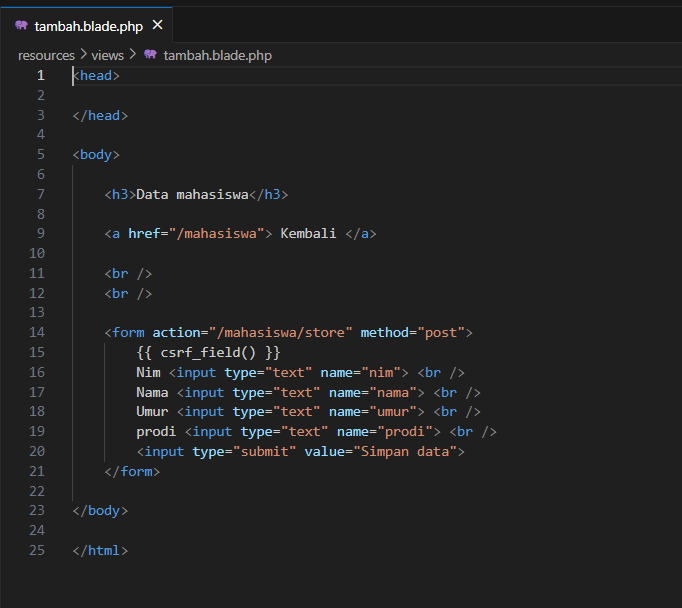
1. Isi file **index.blade.php** dengan potongan kode berikut.



Penjelasan:

Kode diatas berfungsi untuk menampilkan data yang diambil dari tabel mahasiswa yang disimpan dalam variabel $mahasiswa, dimana disini langsung ditampilkan menggunakan perulangan foreach.

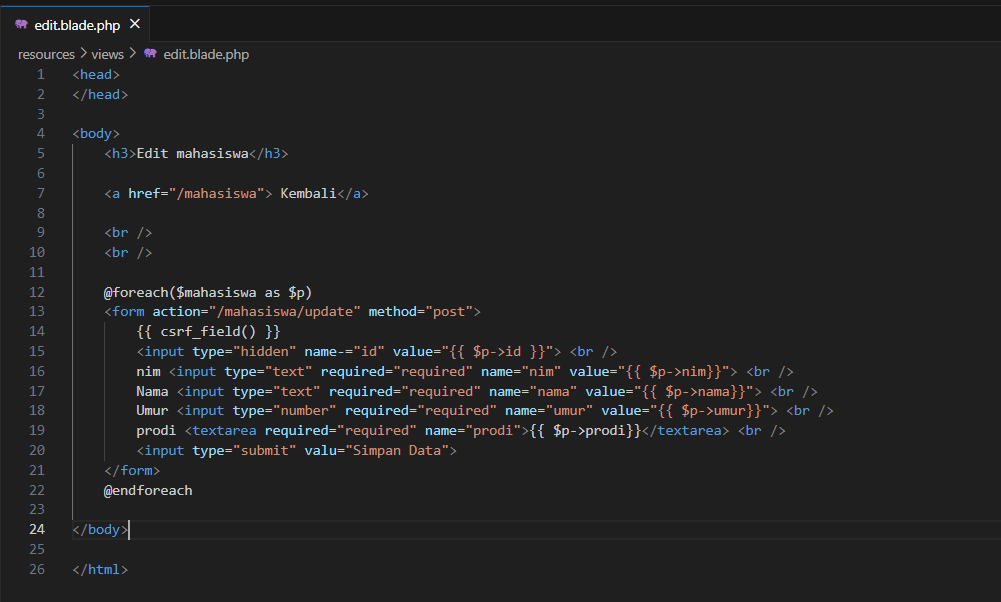
1. Isi file **tambah.blade.php** dengan potongan kode berikut.



Penjelasan:

* Line 17: Action form diarahkan ke route ‘/mahasiswa/store’, karena kita ingin route inilah yang akan menangani pemrosesan data yang di input agar dapat diatur oleh controller.
* Line 28: Merupakan fitur keamanan untuk mencegah penginputan data dari luar aplikasi atau sistem.
* Line 19-23: Merupakan beberapa buah form inputan yaitu Nim, Nama, Umur dan Prodi sesuai dengan format tabel yang sudah dibuat di database sebelumnya.

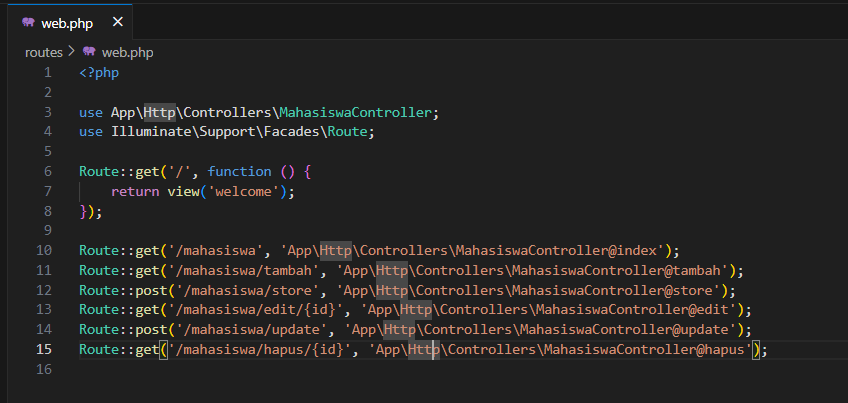
1. Isi file **edit.blade.php** dengan potongan kode berikut.



Penjelasan:

Untuk menampilkan data mahasiswa yang ingin di edit ke dalam form edit digunakan perulangan foreach.

1. Setelah pembuatan controller dan view telah selesai, kemudian lakukan penambahan route pada file **web.php** yang berada pada folder **routes**, sehingga menjadi seperti berikut:



Penjelasan:

* + Line 21: Memerintahkan untuk menjalankan method index() pada controller MahasiswaController pada saat route ‘/mahasiswa’ diakses.
  + Line 22: Memerintahkan untuk menjalankan method tambah() pada controller MahasiswaController pada saat route ‘/mahasiswa/tambah’ diakses.
  + Line 23: Memerintahkan untuk menjalankan method store() pada controller MahasiswaController pada saat route ‘/mahasiswa/store’ diakses. Pada route ini kita menggunakan method **post** karena kita mengirimkan data melalui form ke route ‘/mahasiswa/store’
  + Line 24: Memerintahkan untuk menjalankan method edit() pada controller MahasiswaController pada saat route ‘/mahasiswa/edit/id’ diakses.
  + Line 25: Memerintahkan untuk menjalankan method update() pada controller MahasiswaController pada saat route ‘/mahasiswa/update’ diakses.
  + Line 26: Memerintahkan untuk menjalankan method hapus() pada controller MahasiswaController pada saat route ‘/mahasiswa/hapus/id’ diakses.

1. Setelah semua telah selesai kemudian aktifkan web menggunakan command **php artisan serve**



1. Kemudian kita coba untuk membuka/mengakses website dengan 127.0.0.1:8000/mahasiswa atau localhost:8000/mahasiswa

